Virtual reality applied to a full simulator of electrical sub-stations

G. Romero, J. Maroto, J. Félez , J.M. Cabanellas, M.L. Martínez, A. Carretero

E.T.S. de Ingenieros Industriales, Universidad Politécnica de Madrid, c/ José Gutiérrez Abascal 2, 28006 Madrid, Spain

Abstract

This paper presents an application designed to train electrical sub-station operators by means of a virtual reality environment.

The application allows full viewing of any of the sub-stations in the power supply network. With the appropriate hardware (HMD, 3D mouse and tracking systems) it is possible to navigate into the virtual world and interact with the elements.

Each of the sub-station components has been reproduced in the simulation model, including the behavior laws associated with it, so the complete functionality of the sub-station can be simulated.

This module is built into a larger and more complex computer system composed of the actual sub-station control system, the Geographical Information System which defines the topology of the network, and the functional system which simulates the electrical behavior of the sub-station.

The application automatically updates in the virtual environment any changes to the sub-station's design and allows access, from this environment, to information on every component.

The virtual reality application has been implemented in a hardware configuration and has the same interface as that used in the control system of the real sub-station. In this way, the system developed can be integrated into a replica of the complete power supply network control system emulating a real sub-station, it being able to fully interact with the global system, and allow totally real situations to be simulated.

Keywords: Simulator; Virtual reality; Sub-stations

1. Introduction

In order to properly simulate a virtual world, technologies such as realistic graphics and dynamic simulation with real-time calculations must be used Peripherals must be used for the system to interact with the user. Immersion comes as a result of stimuli to sight, hearing and touch It is possible, therefore, to produce immersion in the system by providing visual, tactile and acoustic feedback to the user.

One of the most common applications of virtual reality lies in simulator development. Simulators can be defined as information systems which reliably reproduce specific phenomena Simulators are mainly used in training although their field of application has grown to include manufacturing and medicine among others.

An additional issue, also related to computing performance, is dynamic simulation. The idea is to reproduce the actual physical behavior by applying the equations governing the simulated system A critical factor is the possibility to solve the equations in real-time, that is, there should be no delay compared to the normal environment's response. There is an important amount of effort being directed to these objectives

This paper deals with the development of an operation simulator for training. The fundamental objective is to develop a simulator for operations at electrical sub-stations. There already exist at least partial implementations of virtual reality based simulators for large installations such as nuclear or petrochemical plants. There are also various applications based on electrical sub-station simulators.

The application has been designed by the Universidad Politécnica de Madrid for Unión Fenosa, one of the main electrical companies in Spain.

2. Description of the application

2.1. Objectives

The fundamental objective is to develop a simulator for operations at electrical sub-stations. These types of operations,

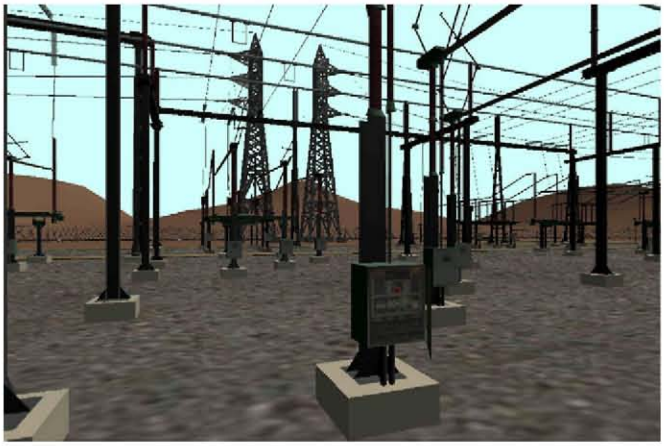

Fig. 1. 3D model of an electrical sub-station.

especially when performed manually or during maintenance, can be considered high-risk activities for the people performing them. Therefore, the use of simulators for training can be particularly beneficial.

Aimed at giving the simulator the highest possible degree of realism, it has been equipped with the following features:

- To be able to closely represent an electrical sub-station. This implies designing three-dimensional geometrical models of all the elements which make up the sub-station (Fig. 1).
- To be interactive. Communication between the Information System and the users must adopt as real a form as possible. Interactivity is obtained by the system responding through the peripherals to user-initiated events.
- To be immersive. For the user to feel he or she is inside the virtual environment, visual and acoustic feedback must be provided through the hardware (helmet with tracking system and sound), with the objects of the environment presented in 3D stereoscopic display.
- The system must replicate, as closely as possible, the actual functioning of the installation. To that end, the functioning logic of the installation has been be coded into the system, so that objects react to user input with the appropriate movement and behavior. Furthermore, all objects must strictly adhere to the physical laws governing their behavior; in this case, the laws of movement affecting three-dimensional objects and the physical principles of electricity which, logically, define the behavior of an electrical sub-station. This means that the system includes mathematical algorithms which simulate the movement of objects simultaneously with the behavior of electrical variables, and that it is able to replicate, for instance, the connection of electrical phases.
- Lastly, the whole system has been integrated into a network, so that it becomes a multi-user system where multiple users can simultaneously input into the same virtual environment, following defined behavioral rules.

These objectives have been attained by interconnecting the different hardware and software elements. The following sections describe how these elements work and their relationship.

2.2. *Previous applications*

The Installations Database (BDI) has been designed to maintain and look-up graphical and textual information on the installations and elements of Unión Fenosa's power supply and telecommunications networks. The information stored in the BDI has been organized on different levels (planning, study, development, operation) together with cartographic information.

The basic functionality of the BDI is as follows:

- Queries: queries against graphical and textual information in the database.
- Maintenance: maintenance of the information in the database.
- Map editing: generation of hard copy and on-screen maps.
- Network analysis: queries based on the topological connections of the network.
- Information exchange: import/export information to/from other systems or official bodies.

The BDI includes the following data:

a. Textual data:
- Every element is uniquely identified through a code.
- Identification and technical data of each installation.

b. Graphical data:
- Cartographic database: rural (communications, hydrographic information, limits, altitude, etc.) and urban (streets, sidewalks, blocks, etc.) maps.
- Detail maps: precise location of the network over a cartographic background.
- Location maps: larger scale representation of the network's location over a cartographic background.
- Schematic drawings: schematic drawings of maneuvers at sub-stations and transformation centers.

The information included in the BDI affords, through the topological connections of its GIS, a full overview of the sub-stations and transformation centers, their internal and external connections and their operational logic (Fig. 2).

2.3. *Technical basis of the application*

To arrive at the functionality indicated in the objectives, several software tools have been used. The core of the application consists of C/C++ code, which accesses the OpenSceneGraph graphical libraries APIs. This aspect of the application allows the user to work with a physical mechanism within a virtual reality environment; that is, to interact with it through devices such as a mouse, stereoscopic glasses, HMDs or gloves.

In the next phase, 3D models of the different components of the sub-station have been constructed. To this end a conventional 3D modeling software has been used. The software allows the insertion of repeated elements as blocks, so that identical geometrical models (i.e., high voltage towers) need not be duplicated. This means a reduction in the resources necessary to

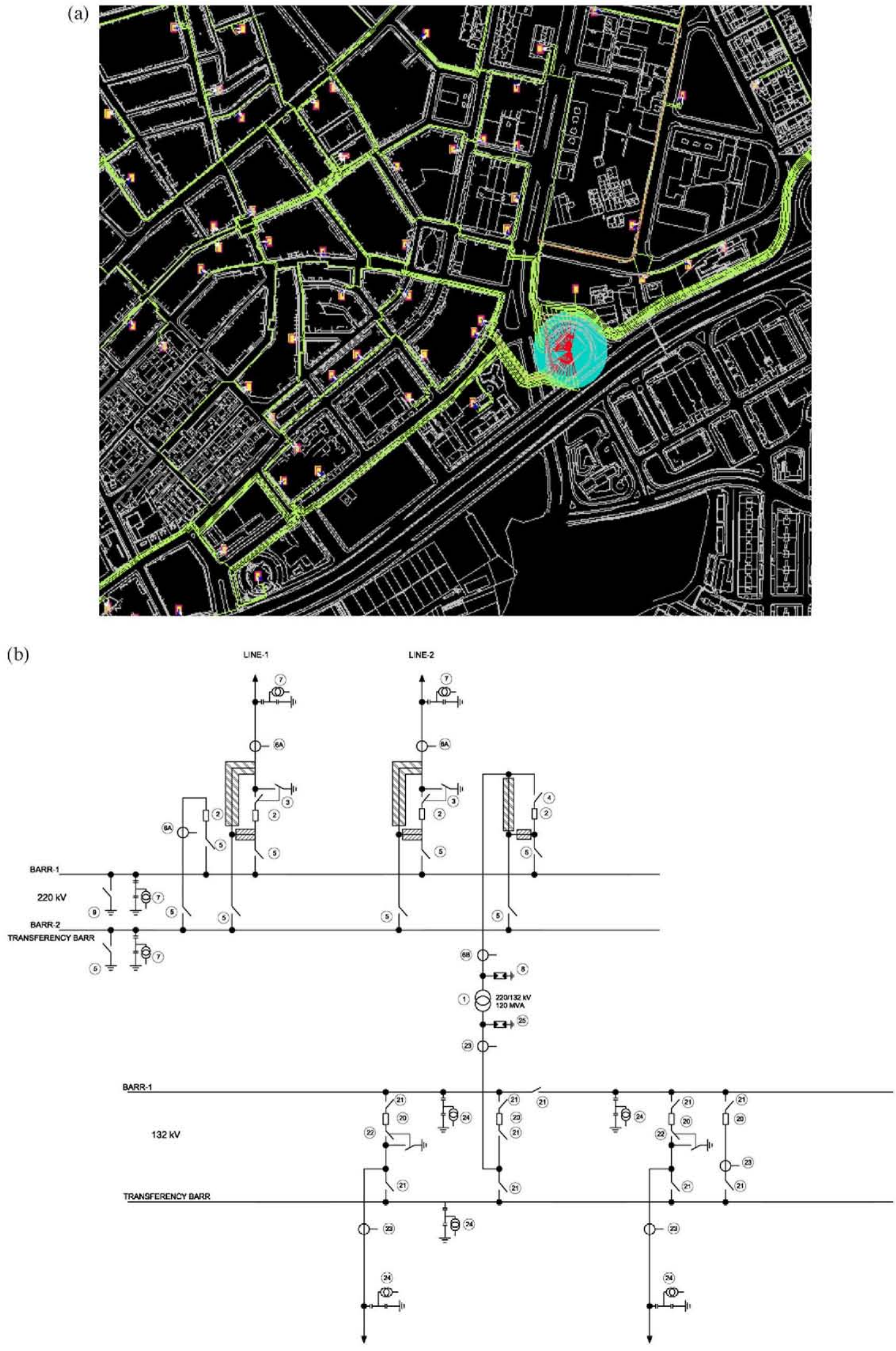

Fig. 2. (a) Geographical database. (b) Schematic drawings of electrical systems.

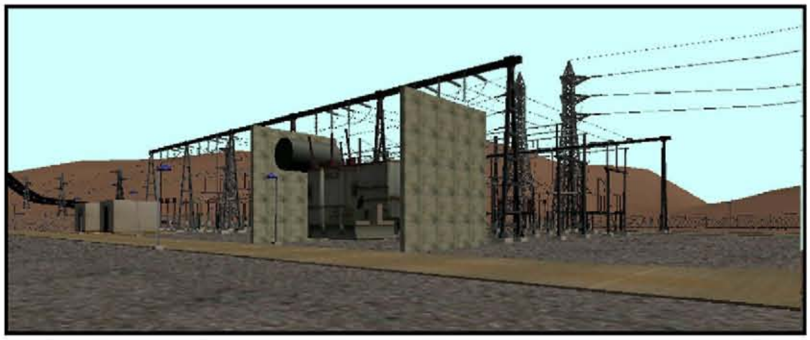

Fig. 3. 3D Geometry and textures.

store the sub-station, in download time from the network, and in processing power to render the sub-station in real-time.

Excessive detail in the geometrical modeling means an increase in the time needed to render it, so that it can become impossible to offer real-time experiencing. The rendering has been optimized by applying textures to the 3D model. This greatly simplifies the rendering without loosing realism (Fig. 3).

Then, the geometrical models are imported by the virtual reality application, which applies to each object properties such as interference and object collision detection, pre-set trajectories and tasks. In order to increase the realism of the whole, colors, transparencies, labels, and lights have been added to the geometries.

Once the geometrical and operational data have been loaded, the virtual environment of the sub-station can be manipulated. Fig. 4 shows the controls of one of the pantographs of the sub-station. When the button on the pantograph's control console is pressed, it moves from open to closed.

In order to view the mechanism in real-time, a scene graph with hierarchical object structure is created. The nodes, that is, the elements which include information on geometry, position or light, are the elements which make up the scene graph. Geometrical, position and light information is stored in geometry, transformation and light nodes, respectively. The nodes are sorted by hierarchy, which means that they are linked vertically and present a tree-like structure. Fig. 5 shows a section of the scene graph corresponding to a sample sub-station.

Depending on the element with which we interact, several different actions have been performed on the virtual sub-station. The sub-station is composed of static physical elements, such as transformers, control elements such as consoles, and assemblies with movement such as pantographs or switches.

- The most general action, which can be applied to every element, is navigation. This consists of interactively changing the viewpoint through the mouse. This is done through what in virtual reality terminology is know as a "motion link" between the computer's input device and the camera's viewpoint. As the input sensor (the mouse, in this case) moves, the viewpoint of the scene changes interactively.
- Operation of mechanisms. There is also a module which performs the kinematic calculations corresponding to show the positions of the parts which make up a mechanism, such as the pantograph in Fig. 6a, so that the moment the system drivers are operated (degrees of freedom) the model will produce a movement following preset kinematic constraints. In order to define the mechanism's kinematic behavior, the system drivers (system input), as well as the kinematic joints making up the system's movement constraints, must be configured. The corresponding scene graph is shown in Fig. 6a.

Fig. 4. Action sequence on the sub-station.

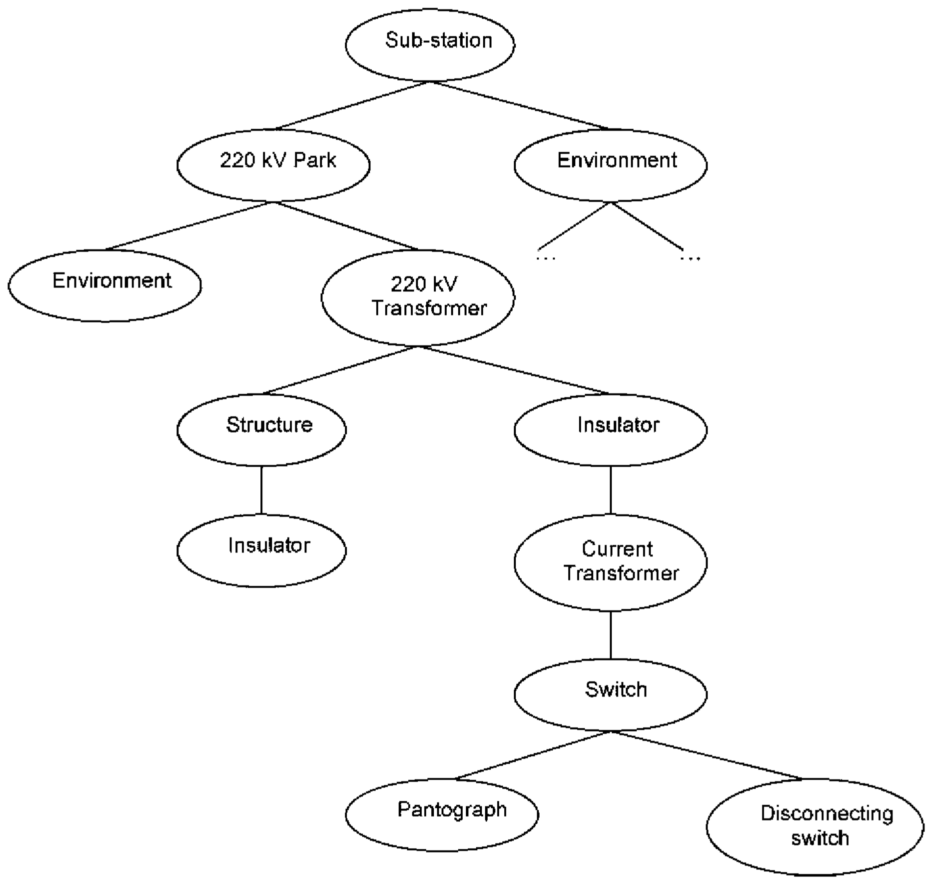

Fig. 5. Scene graph.

- The third type of action is modifying the electrical state: connected, disconnected, grounded, etc. This type of action may or may not be accompanied by a concrete physical movement, but must always be registered and taken into account. In the virtual environment it is indicated by a change in color, for instance. In most cases, electrical maneuvers of this type are subject to operational constraints; i.e., you cannot ground a live element. These constraints are also shown in the scene graph, as in Fig. 6b.

3. Implementation of the project

3.1. Integration of BDI and VR

The Installations Database (BDI) and the virtual reality application (VR) are integrated. The scope of this integration is such that it allows transparent access to information in both systems. Depending on the level of difficulty, integration can be approached differently.

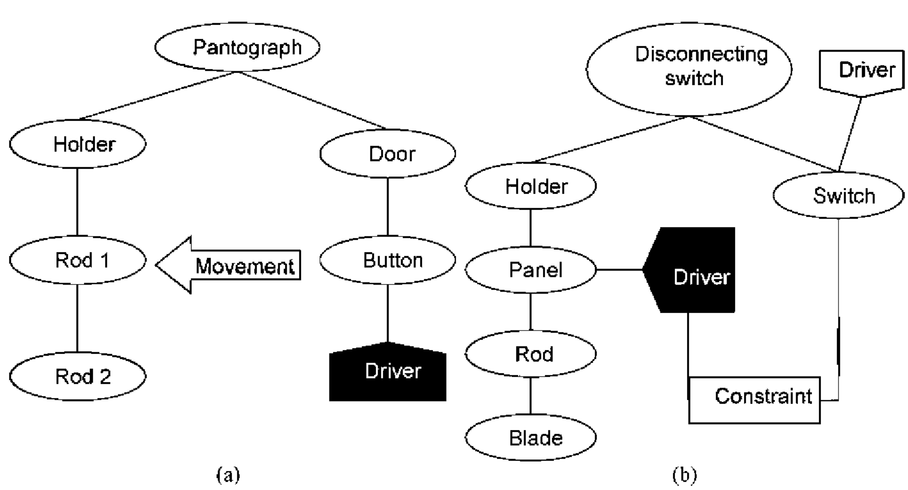

Fig. 6. (a). Scene graph corresponding to the maneuvering of the pantograph. (b) Scene graph with maneuver constraints.

3.1.1. Level I—queries

The following types of queries are allowed:

1. Queries, from the VR application, of textual data corresponding to the modeled elements. A 3D element is selected and the Database Query Application invoked with the identification code of the selected element. A screen is then displayed allowing standard operations of this application (navigating through the hierarchy of installations, locating the element, generating reports, etc.).
2. Accessing the virtual model of a sub-station: a sub-station is displayed using the corresponding textual data contained in the BDI.
3. Locating an element from the BDI: from the BDI, an element can be physically located within the sub-station.

3.1.2. Level II—symbology/behavior

This level allows the user to define, based on specific textual attributes, the symbology and/or dynamic behavior of sub-station elements:

Model: differentiate elements based on model or material.
State: changing the appearance of an element, based on its state (open, closed).
Voltage: same as above, following a live/without voltage criteria.

3.2. Modeling of installations

Simultaneously to the development of the software application, the 3D geometrical models needed by the VR application have been constructed. These models are based on existing drawings, on paper or in digital format, and on actual pictures. The geometric models have been optimized to a high level of realism by optimizing the mesh size, using LOD objects, and applying textures.

The models have been constructed using commercial 3D software, so that the system imports popular or standard commercial formats such as AutoCAD, 3Dstudio, Proengineer, Multigen, IGES, DXF, or VRML. This makes maintenance by the technical department of the company where the application has been installed much easier.

3.3. Physical simulation of the installations

The application developed in the first phase only allows simulation of the sub-station's topology. That is, by manipulating elements, its electrical lines can be connected or disconnected, and this is done by sending information on the state of elements to the Installations Database so that it changes the structure of its topological network.

In cases such as training for maintenance operations, this functionality can suffice. However, in order to go a step further and allow designing or re-designing a sub-station, and work with BDI parameters referring to physical magnitudes governing the operation of the installation, the cross-section of a metal bar for instance, actual behavior laws of the installations need to be introduced.

In this module an application to simulate the physical behavior of the installation has been developed. This simulation method must fulfill several requirements, due to the special character of these installations. Firstly, mechanisms have been simulated in order to represent, within a time framework, the movement of the objects acted upon by the user. Secondly, laws governing electrical phenomena have been reproduced, including magnetic fields. And lastly, integration of all these different behavior laws into a unified simulation environment, interconnecting all fields has also been performed. Additionally, the simulation must take place in real-time.

Lastly, the transitory physical phenomena that can happen in any operation are simulated in real-time using numerical methods. The results of the simulation can be applied to the virtual environment producing a movement, or a special effect, such as sounds, sparks, etc. depending on the phenomena and its intensity in the simulation.

3.4. Distributed interactive simulation

The last phase consists in implementing this system within a Distributed Interactive Simulation (DIS) environment. The objective has been to develop a virtual reality system which meets the specifications of the previous phases, i.e., integration in BDI, realistic behavior laws, and which allows several users simultaneous access to the same installation from different workstations.

This application is based on an Object/Property/Event architecture, and offers the following functionality:

- Standard storage, manipulation and retrieval of objects from a shared database.
- Creation of properties (such as the position of an object) that allow for easy storage of user-defined data (for instance, movement coordinates when an object is moved).
- Triggering of reactions to property changes. A property change is known as an event.
- Property sharing, enabling multi-user simulations.
- The final objective of this phase has been the development of a client/server architecture which allows multi-user, simultaneous generation of interactive graphic simulations.

The software developed is made up of a set of interconnected applications. This solution presents greater scalability if a single application is used that simultaneously takes charge of the sub-station graphic display and the simulation of its logic and behavior.

This scalability allows the implementation of a multi-user environment. Moreover, it gives independence in respect of the power of the computer where the program is being run, since processing can be distributed among different computers. This section describes the methodology implemented, which allows for the interconnection of new behavior modules and the presence of several visuals, thus allowing simultaneous real-time interaction among various users. A distributed environment has been generated made up of the following applications.

3.4.1. Visuals

These are based on OpenSceneGraph This technology gives great flexibility to the software developed, since it allows for a future migration towards operating systems that are different from Windows.

It includes its own programming language (macro language), which enables simple and efficient virtual environments to be generated along with their editing. This language not only allows objects to be inserted, but also contains a set of instructions that enables elements to be inserted, such as atmospheric effects, animated characters, etc. Thanks to this functionality, the user can generate a plain text file, which, together with the 3D geometries, allows any virtual scenario to be reproduced. It allows the loading of geometries generated by graphic design programs and has the capacity to reproduce large scale scenarios with the help of a dynamic load module.

3.4.2. Communications manager

This allows the state of the actuators to be sent from the behavior modules to the visuals, as well as the position and orientation of all the elements in the simulation. It also allows the states of the sensors to be sent from the visuals to the behavior modules.

Its main functions are to interconnect all the applications that form part of the simulation and to manage all the communication flows. Its main feature is to allow the automatic configuration of all the communications from a set of parameters supplied by the user of the software developed. These parameters will define both the policy and the features of these communications. It is based on CORBA [13], which means that applications generated with different programming languages can be integrated.

The following problems have had to be resolved while developing the application:

- Access to variables. A variable cannot be both modified and read at the same moment in time by two threads that are trying to access it simultaneously. Error detection and management. The application must detect errors associated with communications and rectify those capable of rectification.
- Thread management. The communications manager must simultaneously manage data transmission to the display units together with their receipt by the simulators.

Since we are dealing with an application through which all the communications pass, its code is highly optimized. Any loss of performance in the application will affect all the other applications it communicates with, transferring this low performance situation to them. Optimization has been carried out by a meticulous use of dynamic and fixed matrix lists, by minimizing the number of operations present in the algorithms, and selecting and compacting any areas that need to be blocked in order to avoid their simultaneous use by more than one thread. The Communications Manager is based on a protocol that allows communications to be simply configured, it being possible to set their UDP or TCP type. However, as a general rule, one should tend towards UDP data transmissions whenever possible, since this type takes up fewer resources. The Communications Manager has been implemented with CORBA technology. The CORBA components are objects that display services through interfaces that are described in a standard language called Interface Definition Language (IDL), with a similar syntax to that of Java and C++. An IDL definition is then converted, using a language-dependent tool, into one or more files, from which the customer and server, respectively, are coded. In the software developed, an IDL has been defined aimed at being implemented in simulators without the need to modify the different modules comprising the distributed architecture. Therefore, the programmer only has to develop the simulator or set of simulators making up this distributed environment. Moreover, with the help of the interface, the user need only worry about simulating the behaviors, leaving aside managing communications, detecting errors associated with such management, and developing a distributed architecture. The communications manager takes control of all communications by sending the necessary information to each module at an appropriate rate. To this end, each module carries out the following functions:

1. Starting up → the Communications manager assigns a single identifier to each module. Synchronizing → the module clocks and the Manager become synchronized. To obtain good synchronization a maximum error must be set in accordance with the following sequence: Every n milliseconds, send the information from each of the objects controlled.

Fig. 7 shows the general architecture of the system developed.

3.4.3. Behavior modules

A visual without a behavior module allows a scenario to be reproduced at a particular instant. However, if it is wished to reflect the evolution of the environment according to time, and therefore represent the different states that the elements gradually acquire in that environment, a module entrusted to calculate this evolution is needed. In the simulator developed, the logic associated with behavior is introduced by means of a module that allows the interpretation of a set of files containing the behavior of the elements, coded in a language with syntax the same as a PLC.

Thus, a programmable automaton language interpreter for inserting basic behavior has been developed.

The programmable automaton functions in such a way that the outputs depend on the instantaneous value of the inputs. However, the evolution of logic functions of automatism require a specific calculation time. In order to ensure that the input values are not changed during this evaluation, synchronous processing modes are used that only take account of the inputs, and update the outputs in specific instants of time. Their functioning can be summed up as follows:

1. Start of cycle.
- Storage of input values at a particular instant. Running the program; during the entire process, the value of the inputs

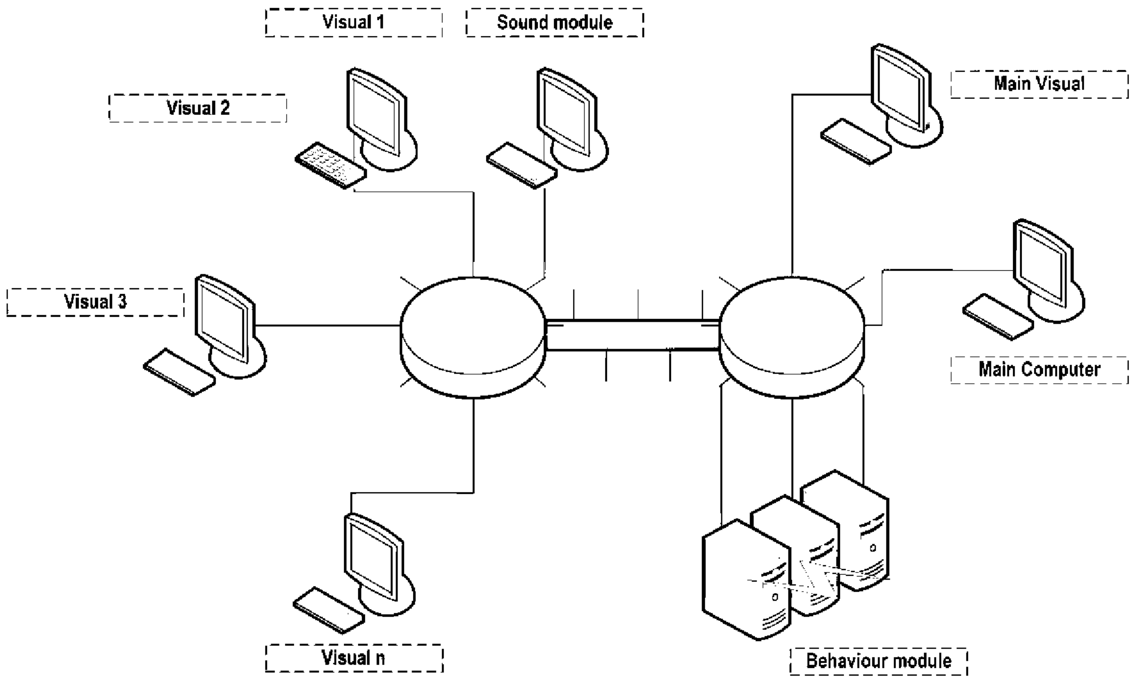

Fig. 7. General architecture of the system.

that is stored remains constant. Simultaneous updating of outputs.
2. End of cycle.
3. Repetition of the process.

The behavior module, therefore, works as an automaton emulator in such a way that with some particular inputs some outputs are generated that are reflected in the environment. In order to generate the variables making up the emulator's outputs, the figure of the sensor has been created inside the visuals, which takes charge of reading the value of a particular property at the start of each automaton cycle.

The actuators have been created in the same way so that the environment can be acted on. These are elements that act on a particular property with the ability to change its value. Both the actuators (behavior module outputs) and the sensors (behavior module inputs) are treated as binary-type variables, that is, their possible values are 0 or 1. Described below are the sensors implemented, their main features and scope of use.

Types of sensors implemented:

- State: This controls whether a visual element is activated or not. It thus allows the user not only to know if a geometry is visible or not, but also if a light is on or off, if a fog-type node is active, etc. Position: This informs if a node is in a position near the sensor. This check is made by means of ranging. Linear position: This detects if a node intersects with the imaginary segment, which, setting out from a point P at the center of a node, has the direction of a vector (x, y, z). The size of the segment is a user-specified parameter. The most typical example of one would be a photoelectric cell. Switch: This behaves like a push-button, that is, it lets current pass only and exclusively during a cycle. At that instant its value is true and then passes to false during the remaining instants even though the button continues to be pressed.
- Button: While the button remains pressed, it lets the current pass taking the true value, passing to false value when the pressing finishes. Movement: This checks the different properties of a movement.

Types of actuators implemented:

- Node visibility control: this lets a node be activated or deactivated, thereby allowing the geometries to be visible or not. If it is a light-type, it switches it on or off, and if a fog-type, it can make it act or not. Variation in the properties of a movement.
- Determining the state of a sound.
- Acting on a Switch element: this allows the child of a switch to be selected each time that it takes the true value or rotate among the various children.

With the help of these sensors and actuators, all the actions needed to manage an electrical sub-station can be generated, opening or closing phases, operating switches, etc.

4. Conclusions

An application designed for training electrical sub-station operators by using a virtual reality application has been set out in this paper.

The application allows full viewing of any of the sub-stations in the power supply network, allowing navigation into the virtual world and interaction with the elements. Each of the sub-station components has been reproduced in the simulation model, including the behavior laws associated with it, so the complete functionality of the sub-station can be simulated.

The virtual reality application has been implemented in such a way that the system developed can be integrated into a replica of the complete power supply network control system emulating a real sub-station, it being able to fully interact with the global system and allow totally real situations to be simulated.

There is no doubt that being able to simulate expensive installations with virtual models which afford the same functionality is an extremely interesting possibility. This virtual reality application is a tool aimed at this interest.

In this complex issue, important technologies and methodologies, such as virtual reality, dynamic simulation, databases, GIS, computer networking, all join together to offer a real-time solution.

References

J. Vince, Virtual Reality Systems, Addisson Wesley, 1995.

M.S. Miller, D.M. Clawson, M.M. Sebrechts, B.A. Knott, Interface design for inducing and assessing immersion in virtual reality, in: Proceedings of the CHI 98 Summary Conference on CHI 98 Summary: Human Factors in Computing Systems, 1998, p. 343.

P.D. Houghton, SMAS: an expert system for configuring a research flight simulator, in: The Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, vol. 2, 1989, p. 601.

P.A. Farrington, B.J. Schroer, J.J. Swain, Y. Feng, Simulators as a tool for rapid manufacturing simulation, in: Proceedings of the 1994 Conference on Winter Simulation, 1994, p. 994.

S. Bayarri, M. Fernandez, M. Perez, Virtual reality for driving simulation, Commun. ACM 39 (May (5)) (1996) 72–76;

G. Singh, S.K. Feiner, D. Thalmann, Virtual reality software and technology, Commun. ACM 39 (May (5)) (1996).

B.J. Schroer, P.A. Farrington, J.J. Swain, D.R. Utley, A generic simulator for modeling manufacturing modules, in: Proceedings of the Conference on Winter Simulation, 1996, p. 1155.

S. Weghorst, Virtual reality applications in health care, in: Proceedings of the CHI 98 Summary Conference on CHI 98 Summary: Human Factors in Computing Systems, 1998, p. 375.

C. Vera, J.A. Lozano, F. Aparicio, J. Félez, SINRAT IV. A program for Traffic accident reconstruction, in: Proc. of the ICBGM'95. SCS. Simulation Series, vol. 27, Number 1, 1995, pp. 209–214.

J. Garcia de Jalón, E. Bayo, Kinematic and Dynamic Simulation of Multibody Systems, Springer-Verlag, New York, 1993.

E. Arroyo, J.L. Los Arcos, SRV: a virtual reality application to electrical substations operation training, in: 1999 IEEE International Conference on Multimedia Computing and Systems (ICMCS'99), vol. 1, 1999, p. 9835.

Y. Mizukami, S. Morita, K. Asano, N. Kamiji, Substation simulation system using virtual reality technology, Electr. Eng. Jpn. 115 (8) (1995) 109–120.

http://www.openscenegraph.org.

http://www.corba.org/.